13. ročník
Žilina 1. – 3. 7. 2025
Otvorený softvér vo vzdelávaní, výskume a v IT riešeniach
ossconf.fri.uniza.sk
OSS Conf

OSSConf 2025: 1–5

Geocaching.com slaví 25 let: GCB78KB, GCB78MA, GC8RXDE

PAVEL STŘÍŽ (CZ)

Abstrakt. Článek připomíná letošní 25. narozeniny geocachingu. Vybral si k tomu kešky vedoucí na problém obchodního cestujícího. Článek zmiňuje klíčové kroky k jejich vyřešení. Autor také předkládá výzvu hackerům, a to jak automaticky řešit puzzle na jigidi.com.

Klíčová slova. TSP, kešky.

GEOCACHING.COM CELEBRATES ITS 25TH BIRTHDAY: GCB78KB, GCB78MA AND GC8RXDE

Abstract. The paper celebrates geocaching.com's 25th birthday! The selected geocaches lead to the Traveling Salesman Problem, the paper provides key steps how to solve them. There is also an open challenge for hackers: how to solve jigidi.com's puzzles automatically.

Keywords. TSP, geocaching.

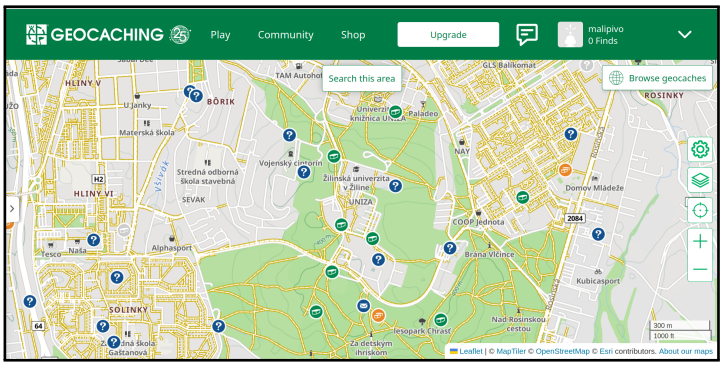
1. O keškování

Pozor, nejsem kačer (hráč geocachingu), byť jsem jich ve skutečnosti pár našel (tzv. odlovil). Okolí mě přimělo několik úloh vyřešit.

Kdybych měl neznalému vysvětlit, o co v keškování jde, tak je to hledání skrytých schránek pomocí GPS navigace. Pro jejich nalezení je často nutné předem vyřešit nějakou úlohu či šifru. Tím získáte GPS souřadnice kešky a můžete vyrazit na lov. Když si tedy vyjdete ven za turistikou, může to být jeden z cílů vašich cest. O svých úspěších si vedete záznamy, můžete se podělit s ostatními, tj. logovat. Lovit chodí jednotlivci, týmy i celé rodiny. Více viz `https://www.geocaching.com/`. V roce 2025 slaví 25. narozeniny. I v Žilině kešky jsou, viz mapa na další straně!

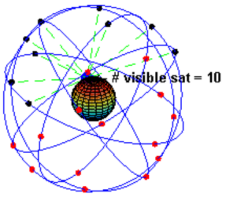
2. Má první vzpomínka na kešky

Přišel za mnou spolužák, kačer Radek Vicherek, že náš učitel a cvičitel z Fakulty technologické ve Zlíně, Vosař (víme o vás, chovateli vos!), připravil řadu kešek, od GPS-1 (GC2MBZP) po Satelity – bonus (GC2JE81), a že s některými nepohne.



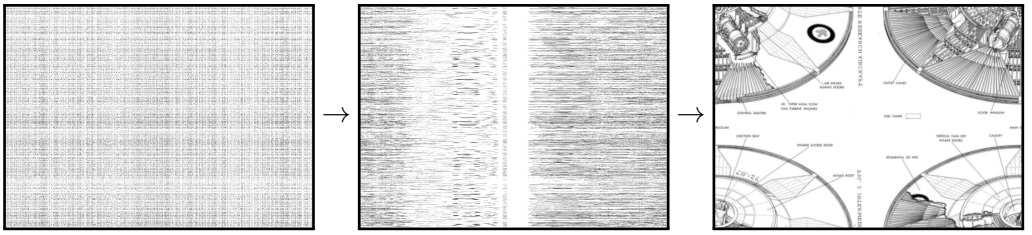
Fajn, říkám si, tak jsem se do nich pustil. Matematicky založené kešky se zvyšující se obtížností. Dozvěděl jsem se zajímavé věci.

Docela mi to odsýpalo, ale u iGPS (GC2JE7V) jsem se na dlouhé měsíce zasekl. Bylo to hodně dané tím, že se stará data špatně sháněla. A až data byla, řešit to přes matematiku je zapeklitá záležitost. Doporučuji! Zapotil jsem se! Ano, to „i“ v iGPS znamená, že musíte být dostatečně chytří, dodávám s úsměvem. Pan učitel se mnou ještě jednou hezky zacvičil! Své postřehy jsem zapsal do logu pod přezdívkou malipivo.



3. Zvířátková 12, 14 a Skartovačka

Několik týdnů před konferencí mě kolega poprosil, jestli bych se nepodíval na tyto: GCB78KB, GCB78MA a GC8RXDE. Problém je opět matematický, máme rastrový obrázek a jsou proházené řádky, případně i sloupce. Úkolem je původní obrázek složit a vyčíst nějaké informace. Konkrétně: zadání, po proházení sloupců a po proházení řádků získáváme řešení (zamíchání se čte zprava doleva):



4. Spoiler Alert!

Celá úloha vede na problém obchodního cestujícího (TSP), kdy matice podobností je nějaká metrika z pixelů příslušných řádků, resp. sloupců. Účelová funkce je minimalizovat zvolenou metriku, jinými slovy se dostat do původního obrázku.

Počet uzlů je počet řádků rastrového obrázku v prvním zamíchání, případně v druhém zamíchání počet sloupců. Konkrétně: Zvířátková 12 je 890 na 579 pixelů, Zvířátková 14 je 871 na 592 pixelů a Skartovačka je 970 na 707 pixelů.

Vedle programu Concorde, viz článek ve sborníku, se svou heuristikou a Domino-Parity je LKH (Lin-Kerninghan-Helsgaun) další možná heuristika. Keška vás navede na Github: image-unshredding od Robina Houstona.

První háček je, že musíte užít parametr `--branch`, abyste měli ve zdrojových kódech to, co potřebujete.

```
git clone --branch double-shuffling
   https://github.com/robinhouston/image-unshredding.git
```

Další problém je, že kód je zapsán v Python2. Tedy buď musíte na přepis do Python3, či přes virtuální prostředí typu MiniConda.

Ale ani to vám nebude stačit. Program LKH v2.0.7 nezkompilujete na první dobrou. Musel jsem jít do poslední verze 2.0.10 z `http://webhotel4.ruc.dk/~keld/research/LKH/`, a to patřičně nasměrovat.

```
wget http://webhotel4.ruc.dk/~keld/research/LKH/LKH-2.0.10.tgz
tar xvf LKH-2.0.10.tgz
cd LKH-2.0.10; make; cd ..
```

Po probádání `Makefile` jsem dílčí řešící kroky usměrnil. Soubor `vstupxx.png` je zadání Skartovačky. `bin/lkh.sh` jsem si zkopíroval na `bin/lkh10.sh` a upravil jeden řádek odkazující na adresář s verzí 2.0.10.

```
malfile="xx" # Volba souboru.
# Proházení sloupců.
bin/compute_scores --cols vstup$malfile.png >vstup$malfile-cols.tsp
bin/lkh10.sh vstup$malfile-cols.tsp vstup$malfile-cols.tour
bin/reconstruct_image.py --cols vstup$malfile-cols.tour
   vstup$malfile.png >vstup$malfile-cols.png
# Proházení řádků.
bin/compute_scores --rows vstup$malfile-cols.png >vstup$malfile-rows.tsp
bin/lkh10.sh vstup$malfile-rows.tsp vstup$malfile-rows.tour
bin/reconstruct_image.py --rows vstup$malfile-rows.tour
   vstup$malfile-cols.png >vstup$malfile-vysledek.png
```

Vězte, že i když tohle všechno víte, nadřete se, poněvadž jsem určitě nějaký krok zapomněl zmínit! Nebudu prozrazovat, jak spustit či si vyčíst z `Makefile` zkompilování `compute_scores.c` či si vyzkoušet simulované žíhání u obrázků Nayuki: úkolem je nahodit si Javu v65.0.

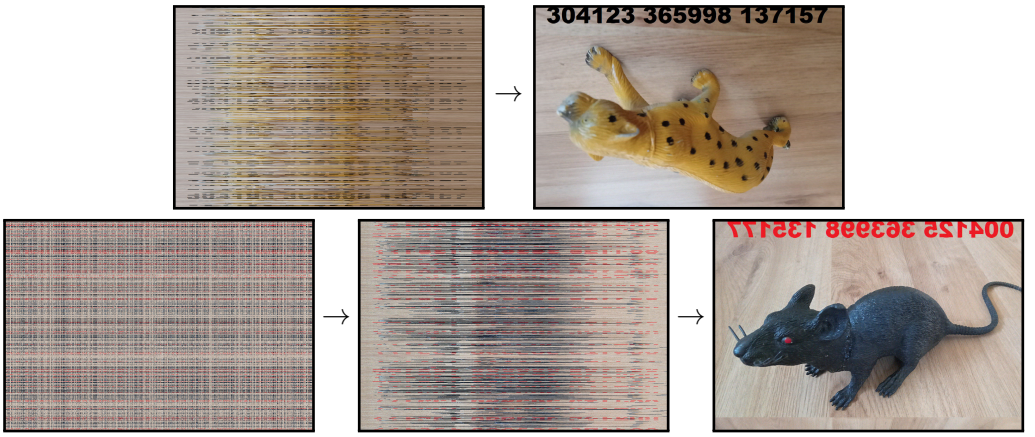
Pozn. I když jsem použil virtuální prostředí pro Python2

```
conda create -n python2p7 python=2.7
source activate python2p7
```

program si natahoval knihovnu `random` z knihoven Python3. Tedy zadání nevygeneruji v době psaní tohoto článku, pro kešku to nebylo třeba.

Měl jsem tak trochu dojem, že jsem se ocitl v reálném programovacím světě! Už jen to, že vstupní .jpeg soubory musíte exportovat do .png, aby se chytla knihovna `Pillow` pod Pythonem, je drobnost, která vás donutí nahlédnout do zdrojových kódů.

V každém případě se mohu pochlubit řešeními Zvířátkové 12 a 14.



Tři šestimístná čísla jsou zakódované souřadnice. Říkají tomu Reverse Wherigo (wherigo je typ keší). Existuje na to online dekodér, `http://www.geekfrog.ca/WIG/ReverseWigDecode.aspx`. Skartovačka obsahuje po sestavení obrázku ještě šifru.

5. Otevřený problém?

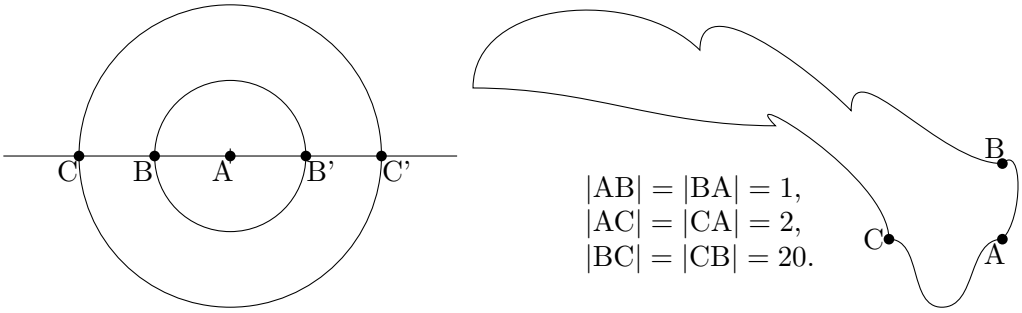
Na chvíli jsem se zamyslel, jestli obecnou matici podobností/vzdáleností lze vykreslit do 2D ve formě bodů. Mohu si rovnou odpovědět, že nikoliv. Zde je ukázka pro tři body A, B a C. Máme matici vzdáleností, symetrickou podle hlavní diagonály, s nulami na hlavní diagonále.

$$\begin{pmatrix} 0 & 1 & 2 \\ 1 & 0 & 20 \\ 2 & 20 & 0 \end{pmatrix}$$

Pokud si zvolíme bod A a dokreslíme bod B, kružnice o poloměru 1, a bod C, kružnice o poloměru 2, obě se středem v A. Zjistíme, že mezi body B a C nelze získat vzdálenost 20, tedy kružnici o poloměru 20, bez ohledu na to, kde si bod B a C na příslušné kružnici vybereme. Nejvíc lze získat tři jednotky, viz |BC'| nebo |B'C|. Nejméně by šla jedna jednotka, viz |BC| nebo |B'C'|. Nakreslit to neumím, ale řešit TSP lze.

Jistá možnost by byla neužít úsečky, ale nějakou křivku ve stylu hranic na reálné mapě. To by byla skutečná výzva, jak to zautomatizovat, kdyby to bylo užitečné! Dává to možnost užít různé typy křivek, fraktálů apod. Mohlo by to být i esteticky zajímavé, co zvolit. V pravém obrázku naznačím náčrtek, který

ukázkovou matici vzdáleností splňuje, vlevo tomu tak není. Těžko odhadovat, jestli by to pomohlo s vizualizací map vah koeficientů u umělých neuronových sítí, což je aktuální téma (AI).



6. Otevřená výzva hackerům!

Nedávno mě kolega poprosil, jestli bych neudělal automatického řešitele na `https://www.jigidi.com/`, že u řady kešek se musí odehrát puzzle, a že to není pro něj. S tímto jsem zatím nepohnul, ale osobně se rád podělím, jak si na tom stojím. Přesněji se tedy chce, aby program ve webovém prohlížeči poskládal puzzle. Nebo to jakkoliv obešel. Po složení puzzle se zpravidla objeví zpráva, která pomůže s keškou.

Buď hackerem a zkus to nějak vymyslet! Kdyby se podařilo, napište mi, kačerům ze svého okolí bych výsledek předal. Pro testování lze použít např. `https://www.jigidi.com/solve/92zdkwnh/1-10/` (GCAQJ2E).

Za inspirativní považuji `https://danq.me/2021/08/26/jigidi-solver/`, byť by to v současnosti neměla být použitelná cesta, dle slov autora. Také existují databáze řešení takových puzzle ve vztahu ke keškám, viz např. `https://solvedjigidi.com/`, kde si lidé sdílejí řešení.

7. Závěrem

Je $\mathcal{P} = \mathcal{NP}$? To já zatím nevím, dodávám s úsměvem, vždycky jsem si myslel, že v žádném případě, ale TSP a vývoj heuristických algoritmů plus rozepsání libovolného problému v třídě $\mathcal{NP}$ pro SAT řešitele dává jistou naději, že jsou si rovny. A jestli k důkazu pomůže nějaká keška, proč ne, že?

Kontaktní adresa

Ing. Pavel Stříž, Ph.D., Nakladatelství Martin Stříž, U Škol 940, Bučovice, okres Vyškov, 685 01, Česká republika,
E-mailová adresa: `pavel@striz.cz`